Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6155925, 8 pages
https://doi.org/10.1155/2022/6155925

Research Article

Deep Learning-Based Anomaly Traffic Detection Method in Cloud Computing Environment

Junjie Cen[1] **and Yongbo Li**[2]

[1]*College of Computer Science and Technology, Henan Institute of Technology, Xinxiang, Henan 453002, China*
[2]*College of Computer and Information Engineering, Henan Normal University, Xinxiang, Henan 453002, China*

Correspondence should be addressed to Junjie Cen; cen@hait.edu.cn

Received 25 January 2022; Revised 3 March 2022; Accepted 7 March 2022; Published 31 March 2022

Academic Editor: Shalli Rani

To address the problem of poor detection performance of existing intrusion detection methods in the environment of high-dimensional massive data with uneven class distribution, a deep learning-based anomaly traffic detection method in cloud computing environment is proposed. First, the fuzzy *C*-means (FCM) algorithm is introduced and is combined with the general regression neural network (GRNN) to cluster the samples to be classified in the original space by FCM. Then, the GRNN model is trained and the center point is updated using the sample closest to the FCM clustering center until a stable cluster center is obtained. The parameters in FCM-GRNN are optimized using the global optimization feature of the modified fruit fly optimization algorithm (MFOA), and the optimal spread value is found using the three-dimensional search method through an iterative search. Finally, experiments are conducted based on the KDD CUP99 dataset, and the results demonstrate that the detection rate (DR) and false alarm rate (FAR) of the proposed FCM-MFOA-GRNN method are 91% and 1.176%, respectively, which are better than those of the comparison methods. Therefore, the proposed method has good anomaly traffic detection ability.

1. Introduction

Nowadays, network traffic anomaly detection has become an important part of cyberspace security, and the explosive growth of traffic data has led to increasing requirements for efficiency and robustness when various methods are applied to learn data [1]. For example, the update of communication protocols and hardware upgrades have a great impact on the stability of the whole network environment [2, 3]. On the other hand, the scenarios of network attacks and the corresponding means of attack have become much more complex, and the currently used traffic anomaly detection techniques are likely to be no longer applicable at some point in the future. The development of traffic anomaly detection should always be one step ahead of the attackers, especially when the current techniques are already relatively mature and well known to the attackers. It is necessary to open up new research directions [4–7]. In recent years, research on deep learning models has mainly focused on the fields of speech, image, and natural language and has received more and more attention because of the outstanding achievements [8–11].

Cloud computing can provide users with various resources in the form of services through the network. "Everything can be a kind of service and can be provided to users in the form of lease" is the basic concept of cloud computing [12–14]. However, the rapid development and the universal application of cloud computing brings some new problems which cannot be underestimated. The first and foremost problem is the security of cloud computing, which is increasingly widely concerned by the industry [15–17]. Cloud computing has many features, such as self-service on demand, Internet access, fast elastic architecture, virtualized resource pools, measurability, and multiuser. Although these features provide a more convenient and faster computing mode to users, they also pose new challenges to the security of cloud computing platforms.

To address the problem of poor detection performance of existing intrusion detection methods in the environment of high-dimensional massive data with uneven class

distribution, a deep learning-based anomaly traffic detection method in cloud computing environment is proposed. The contributions are as follows:

(1) The fuzzy *C*-means (FCM) algorithm is introduced and combined with the general regression neural network (GRNN). The samples to be classified in the original space are clustered by the FCM algorithm, and the sample closest to the FCM clustering center is used to train the GRNN model and update the center until a stable clustering center is obtained, which improves the stability of the anomaly traffic detection system

(2) The parameters of the FCM-GRNN method are optimized by using the global search feature of the modified fruit fly optimization algorithm (MFOA). And the optimal spread value is found by an iterative search using the three-dimensional search method with the keen olfactory and visual functions of fruit flies, so that the proposed algorithm can converge faster

2. Related Works

In recent years, scholars have conducted in-depth research on abnormal traffic detection methods. The results show that for all abnormal traffic detection data sets, deep learning methods are better than traditional methods. Literature [18] proposed a sliding window abnormal traffic detection method based on the mixed dimension of time and space. The detection algorithm adopts the combination of machine learning and neural network. A sliding window anomaly detection method based on network traffic was studied in the Literature [19]. The method combined the sliding window and deep learning architecture to analyze network traffic, and features in each window were extracted, vectored and then put into a deep neural network for training. Literature [20] proposed a network intrusion detection method based on a lenet5 model, which improved the detection accuracy. Blanco et al. used the genetic algorithm (GA) to optimize a CNN classifier to find better input feature combination [21]. Literature [22] converts variable length data sequence into fixed length data through LSTM and uses an automatic encoder to process fixed length data under unsupervised conditions, so as to reduce the dimension of input data and extract reliable features at the same time. On the basis of cross validation, the threshold is set to classify the abnormal parts in the input traffic data series. In Literature [23], a deep autoencoder-based intrusion detection method was investigated with layer-by-layer greedy training to avoid overfitting. A self-learning framework based on stacked self-encoders for feature learning and dimensionality reduction was proposed in Literature [24]. It applied the support vector machine (SVM) approach for classification, which shows good performance in two-class and multiclass classification. In Literature [25], an unsupervised deep autoencoder model was used for training so as to learn normal network behaviors and generate optimal parameters. Then, the estimation algorithm of the ADE model was introduced in a supervised deep neural network model to efficiently tune its parameters and classify the network traffic. Literature [26] proposed a supervised LSTM-based intrusion detection algorithm that can detect DoS attacks and probe attacks that have unique time series features. Zhang et al. proposed a parallel cross convolution neural network (PCCN) based on deep learning [27]. By fusing the traffic features learned from the two branches of CNN, a better feature extraction effect is obtained. Literature [28] combines CNN and LSTM to learn the temporal and spatial characteristics of network traffic. The above methods are difficult to effectively mine data features and have poor detection performance in the face of high-dimensional data, resulting in low detection rate as well as high false alarm rate.

3. Application Scenarios of the Proposed Method

In the design process of the anomaly traffic detection and analysis model, the principle of modular design is followed. The modular design of the anomaly traffic detection and analysis is conducive to simplifying the complex problems, which is easy to find the problem in the design and can facilitate the update and maintenance of the system at a later stage. The specific functions of each module are shown as follows. As shown in Figure 1, the whole model can be divided into four major modules: SDN controller module, traffic collection module, traffic analysis module and traffic cleaning module.

SDN controller can realize the centralized control of the whole network. The floodlight controller is used to divert the traffic from each OpenFlow switch to the traffic collection module to collect network traffic. As illustrated in Figure 1, the traffic in switch A, switch B, and switch C will be controlled by the SDN controller and converged to the traffic collection module through the secure channel. The traffic analysis module is the core of the entire anomaly traffic detection model. It uses the FCM-MFOA-GRNN algorithm to cluster and analyze the collected traffic to separate the normal traffic from the attack traffic with different attack behaviors. The traffic cleaning module consists of many physical devices that can clean different attack traffic, such as IDS, UTM, WAF, and other physical devices.

4. The Proposed Method

4.1. Algorithm Flow Chart. Although the FCM algorithm can cluster the data and perform mining analysis, many intrusion ways cannot be accurately classified because there are many kinds of data characterizing intrusion categories in intrusion detection systems and the differences between these data are subtle. Therefore, combined with the characteristics of GRNN, this paper proposes an improved FCM-MFOA-GRNN algorithm based on the FCM algorithm. The flow chart of FCM-MFOA-GRNN algorithm is shown in Figure 2. It can be seen that the core module of the algorithm includes five parts, which are the FCM clustering algorithm, initial selection of network training data, MFOA-GRNN network training, MFOA-GRNN network prediction, and network training data selection in order.

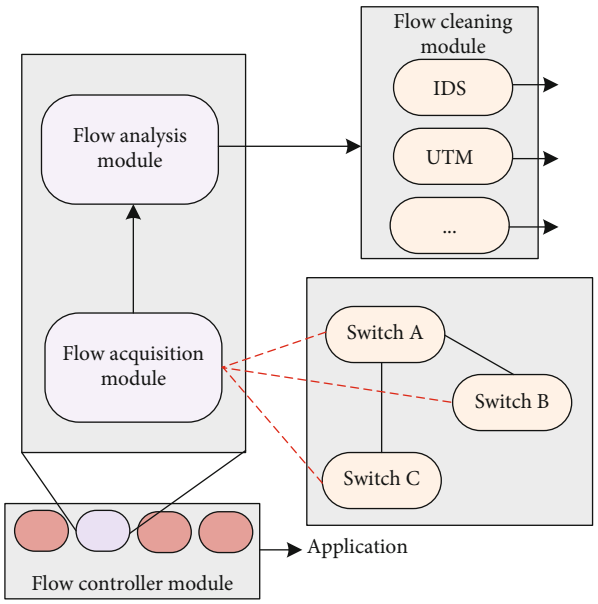

Figure 1: Anomaly traffic detection model based on FCM-MFOA-GRNN.

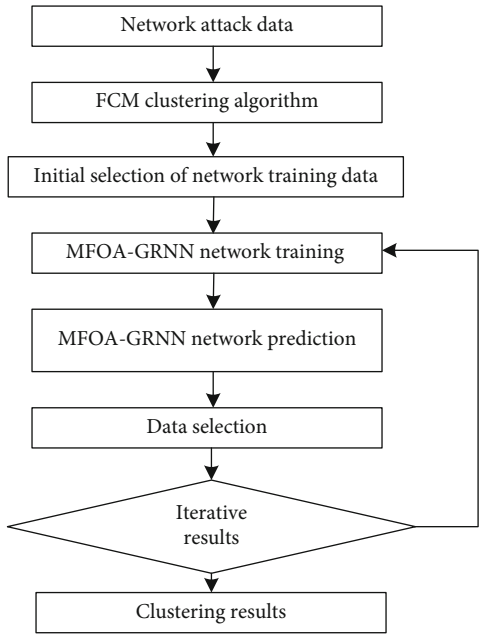

Figure 2: Flow chart of the FCM-MFOA-GRNN algorithm.

4.2. MFOA-GRNN Network

4.2.1. Network Structure of GRNN. Figure 3 shows the structure diagram of the GRNN network. The input of the network is $X = [x_1, x_2, \cdots, x_n]^T$, the output is $Y = [y_1, y_2, \cdots, y_n]^T$.

(1) The number of neurons in the input layer is equal to the vector dimension of the learning sample and is the same as the number of neurons in the mode layer. The neuron transfer function in the mode layer is

$$p_i = e^{\left[-(X-X_i)^T(X-X_i)/2\sigma^2\right]}, \quad (1)$$

where X is the network input variable and X_i is the learning sample corresponding to the ith neuron.

The input of neurons is

$$D_i^2 = (X - X_i)^T(X - X_i). \quad (2)$$

(2) There are two types of summation applied in the summation layer; the first one is

$$\sum_{i=1}^{n} e^{\left[-D_i^2/2\sigma^2\right]}. \quad (3)$$

It performs arithmetic summation on the outputs of all neurons in the mode layer, and the transfer function can be written as

$$S_D = \sum_{i=1}^{n} P_i. \quad (4)$$

Another calculation formula is

$$\sum_{i=1}^{n} Y_i e^{\left[-D_i^2/2\sigma^2\right]} \quad (5)$$

It performs a weighted summation of all neurons in the mode layer, and the connection weight between the ith neuron in the mode layer and the jth neuron in the summation layer is the jth element of the ith output sample Y_i. Thus, the transfer function can be formulated as

$$S_{N_j} = \sum_{i=1}^{n} y_{ij} P_i. \quad (6)$$

(3) The output of the neuron corresponds to the jth element of the estimation result, and the output can be written as

$$y_i = \frac{S_{Nj}}{S_D}. \quad (7)$$

4.2.2. Network Flow of MFOA-Optimized GRNN. The performance of GRNN can be directly affected by the value of σ. This paper proposes a new MFOA-optimized GRNN, which is named as MFOA-GRNN, for the purpose of optimizing the spread value. FOA is prone to local extremes and cannot search for the global optimum, which is mainly caused by its fitness function. Hence, the fitness function must be modified to get rid of the local extremes. On the other hand, if the distance Dist(i) is positive, its reciprocal

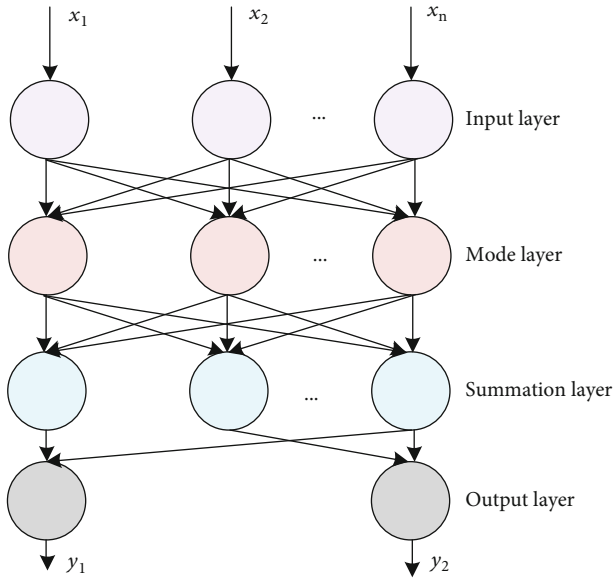

Figure 3: Structure of GRNN.

must be positive, which is the lack of negative values of fitness function as pointed out by many scholars. By adding S_i to an escape parameter, not only can the local minima be got rid of but also the fitness function can get negative values.

In addition, the flight area of fruit flies in real life is three-dimensional space, which is different from the two-dimensional search space of the original fruit fly algorithm. Using the three-dimensional space search method, the optimal spread value can be found iteratively by using the sharp olfactory and visual abilities of fruit flies. At this point, the mean squared error (MSE) is the smallest and σ is the optimal concentration value for taste. The foraging diagram of fruit flies in the three-dimensional space is illustrated in Figure 4.

The specific implementation steps are as follows.

Step 1. Randomly generate the initial position $(X_{\text{Init}}, Y_{\text{Init}}, Z_{\text{Init}})$, the number of individuals, and the maximum number of iterations of the fruit fly group.

Step 2. Define random flight direction and distance:

$$\begin{aligned} x_i &= X_{\text{Init}} + \text{random value}, \\ y_i &= Y_{\text{Init}} + \text{random value}, \\ z_i &= Z_{\text{Init}} + \text{random value}. \end{aligned} \tag{8}$$

Step 3. Calculate the distance $\text{Dist}(i)$ between each point and the initial point.

$$\text{Dist}(i) = \sqrt{\left(x_i^2 + y_i^2 + z_i^2\right)}, \tag{9}$$

$$S_i = \frac{1}{\text{Dist}(i)} + \Delta, \tag{10}$$

where S_i is the distribution parameter of GRNN.

Step 4. The mean square error is used as the determination function of taste concentration:

$$\text{Smell}(i) = \text{MSE}(i) = \frac{1}{n}\sum\left(y_{\text{pre}} - y\right). \tag{11}$$

Step 5. Find the individual with the optimal $\text{Smell}(i)$ in the population, i.e., the minimal value of MSE.

Step 6. Retain the optimal taste concentration value S_i and the corresponding coordinate; the population will fly towards that position using visual advantage.

Step 7. Repeat steps 2 to 5 to repeatedly find the best solution. If true, proceed to step 6.

Step 8. Determine whether the maximum number of iterations is reached, and take the optimal spread value retained into the GRNN model to obtain the final prediction result.

4.3. Intrusion Detection Model Based on the FCM-MFOA-GRNN Algorithm. The intrusion detection model based on FCM-MFOA-GRNN algorithm consists of the FCM clustering algorithm, initial selection of network training data, MFOA-GRNN network training, MFOA-GRNN network prediction, and network training data selection.

(1) The role of the FCM clustering algorithm is that when a large number of network attack data streams drained from the software-defined network enter into the system, these data streams are preprocessed first and then divided into n classes using the FCM algorithm, in which the clustering center c_i and affiliation matrix U of each class can be obtained

(2) The selection of the initial data of network training is based on the selection of those samples closest to each type of center from the results of FCM clustering as the initial data of network training:

(3) Randomly generate the initial position $(X_{\text{Init}}, Y_{\text{Init}}, Z_{\text{Init}})$, number of individuals, and maximum number of iterations of the Drosophila population and generate the random direction and distance of flight

Step 1. First find the sample mean mean_i of each class in the n classes divided from the FCM clustering separately.

Step 2. Then, for all samples X in each class, calculate their distances to the sample mean mean_i separately to form a distance matrix d_i.

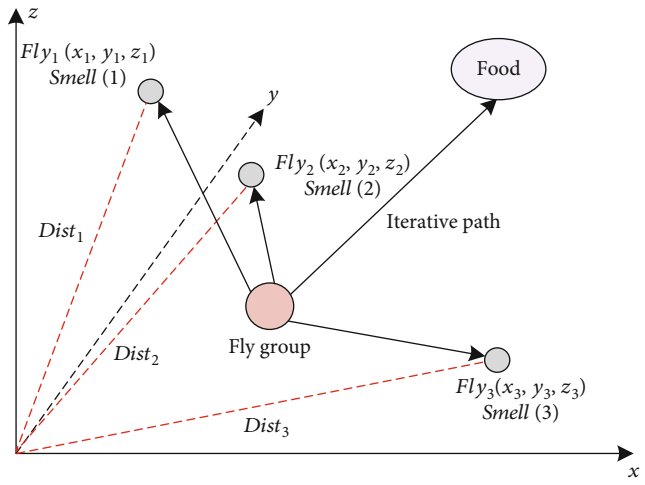

FIGURE 4: Schematic diagram of the iterative search for food of the fruit fly group.

Step 3. Find m number of samples with the shortest distance in the matrix d_i, and compose them into a group. Assume that their corresponding outputs are i. As n classes are divided finally, a total of $n \times m$ groups of training data can be obtained after this step. At this point, the network intrusion feature vector is the input, and the intrusion class is its output.

Calculate the distance Dist(i) between each point and the initial point and the taste concentration determination value S_i. Let S_i be the distribution parameters of GRNN.

(4) MFOA-GRNN network training. The role of this section is to take the selected training data to train the FOA-GRNN network. This is done in MATLAB by using the GRNN network training function newgrnn()

(5) Based on all input sample data X, the network output sequence Y corresponding to them can be predicted

4.4. Evaluation Metrics. The algorithm performance is demonstrated by metrics such as the detection rate (DR) and false alarm rate (FAR). DR refers to the percentage of the number of abnormal data correctly detected in the actual number of abnormal data, which reflects the probability that an attack will be detected. FAR refers to the percentage of the number of abnormal data incorrectly detected in the number of all detected abnormal data, which reflects the probability of a normal behavior being treated as an attack. These two metrics are calculated as

$$\mathrm{DR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{12}$$

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}, \tag{13}$$

where TP is the abnormal data detected as abnormal, FN is the abnormal data detected as normal, TN is the normal data detected as normal, and FP is the normal data detected as abnormal.

5. Experiment and Analysis

5.1. Hardware. The experimental platform is based on Windows 8, configured with Intel(R) Core(TM) i7 CPU M370 @ 2.7 GHz, 16 G memory, 500 G hard disk. The simulation is conducted on MATLAB R201b6 and neural network toolbox. Hadoop is used to build a cloud computing platform. Hadoop is an open-source distributed data processing framework and contains the functions needed for cloud computing, mainly including distributed file system HDFS, distributed computing model MapReduce, unstructured file storage system HBase, relational database transfer tool Sqoop, and distributed cluster negotiation service software Zookeeper.

5.2. Dataset and Preprocessing. In this paper, the algorithm is trained using the KDD CUP99 dataset with 500,000 training subsets and 500,000 test subsets, using the training subsets as the data for the training section of the algorithm. In the 500,000 training subsets, there are 97,278 pieces of Normal, 391,458 pieces of DoS, 52 pieces of U2R, 1,126 pieces of R2L, and 4,107 pieces of Probe. The KDD CUP99 dataset is processed network traffic data with 41 features for each connection, including 9 basic TCP features, 13 content features of TCP connections, 9 time-based network traffic statistics features, and 10 host-based network traffic statistics features.

These features are character-based and numeric, where numeric features contain discrete numbers and continuous numbers. And the data range of each feature varies greatly, which would make the features with low order of magnitude lose information if the raw data is used directly. In order to improve the accuracy of the machine learning algorithm, the dataset is standardized and normalized, which is processed as follows.

(1) Numerical processing: convert character-based features to numerical features

(2) Standardization: first, the mean value of each attribute and mean absolute error can be calculated as

$$\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{ik}, \tag{14}$$

$$S_k = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (x_{ik} - \bar{x}_k)^2}, \tag{15}$$

where $\bar{x}_k$ denotes the mean value of the kth attribute, S_k denotes the mean absolute error of the kth attribute, and x_{ik} denotes the kth attribute of the ith record.

Next, each data record is standardized, which can be calculated as

$$Z_{ik} = \frac{x_{ik} - \bar{x}_k}{S_k}, \tag{16}$$

where Z_{ik} represents the value of the kth attribute in the ith data record after normalization.

(3) Normalization: normalize each value after standardization to the interval [0,1].

$$X' = \frac{X - \min}{\max - \min}, \tag{17}$$

where max and min are the maximum and minimum values of the sample data, respectively.

5.3. Iterative Optimization Trajectory of the Proposed Method. Let the initial position of Drosophila population be $[0, 0.5, 0]$, the population size be 8, and the number of iterations be 150. Select 200 groups as training samples and 10 groups as prediction samples. The models proposed in this paper are used for prediction at the same time, and the results are shown in Figure 5. It can be seen that the fruit fly group in the proposed model does not follow a certain directional path to find the optimal solution sequentially, but there are only 6 position points in the trajectory route.

5.4. Intrusion Detection Results Based on the FCM-MFOA-GRNN Algorithm. In order to reflect a real network environment as much as possible, a number of data are selected from the KDD CUP99 dataset to create 5 groups of datasets, each of which contains 3800 normal data and 100 attack data. And these 5 groups of datasets need to be as even as possible in selecting attack categories. Table 1 shows the results of three simulation experiments on each dataset using the FCM-MFOA-GRNN algorithm, respectively, and the experimental results are taken as the average of the three results. Among them, two parameters are important metrics that can indicate the performance of the algorithm, i.e., DR and FAR. As shown in Table 1, DR and FAR of the proposed method are 91% and 1.176%, respectively.

In order to demonstrate the performance of the proposed method, it is compared with the methods proposed in Literature [27] and Literature [28] under the same experimental conditions, and the comparison results are shown in Table 2. From the experimental results, it can be noted that DR of the method proposed in Literature [27] is only 89.24% and FAR is 2.075%. DR of the method of Literature [28] is 90.1% and FAR is 1.237%. DR and FAR of the proposed method are 91% and 1.176%, respectively, which are better than those of the comparison methods. This is because the proposed method combines the FCM algorithm with GRNN, so as to cluster the samples to be classified in the original space by the FCM algorithm and then use the sample closest to the FCM clustering center to train the GRNN model and update the center point until a stable clustering center is obtained. Therefore, it can better distinguish the small differences of attributes in complex spatial data and improve the accuracy of detection. In contrast, the comparison methods do not effectively mine the features of high-dimensional data, and therefore, the detection performance is poor.

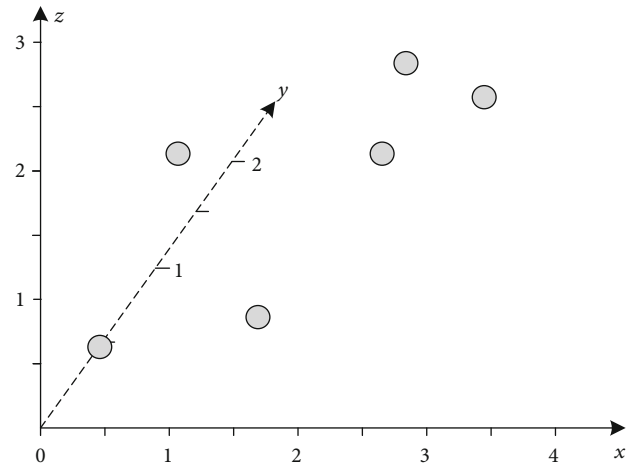

Figure 5: Iterative optimization trajectory of fruit flies in the proposed method.

Table 1: Intrusion detection results based on the FCM-MFOA-GRNN method.

	Normal	Attack	DR	FAR
Date set 1	3742	91	91%	1.175%
Date set 2	3758	93	93%	1.297%
Date set 3	3756	90	90%	1.138%
Date set 4	3697	89	89%	1.109%
Date set 5	3762	92	92%	1.161%
Average value	3743	91	91%	1.176%

Table 2: Performance comparison of different methods in intrusion detection.

Method	DR	FAR
Literature [27]	89.24%	2.075%
Literature [28]	90.1%	1.237%
Proposed method	91.0%	1.176%

In order to compare the running time of the proposed method with the methods of Literature [27] and Literature [28], different amount of data are selected from the KDD CUP99 dataset for testing. The running time of each algorithm was compared, and the results are shown in Figure 6. The minimum number of data selected is 200, and the maximum number is 20000. It can be easily seen from Figure 6 that the method of Literature [27] takes the most average detection time during the whole experiment, and the proposed method takes the shortest time. This is because the parameters of the FCM-GRNN model are optimized by using the global search feature of MFOA, and the three-dimensional search method is used to find the optimal

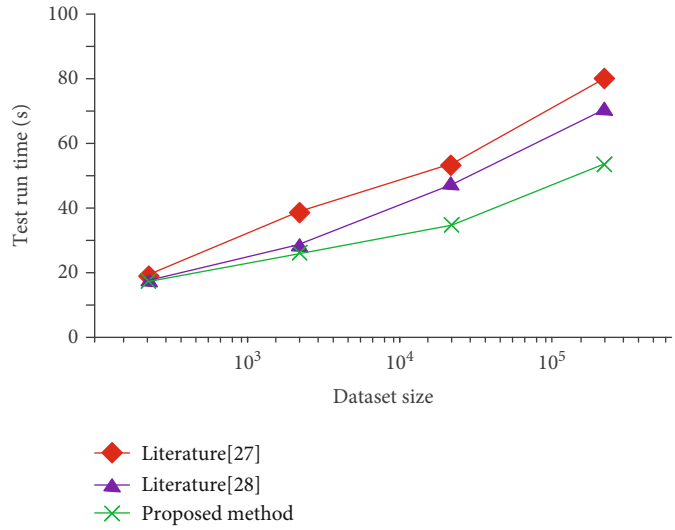

Figure 6: Comparison of running time when different sizes of datasets are detected by different methods.

spread value iteratively by using the sharp olfactory and visual advantages of fruit flies, making the algorithm converge faster and the time consumption be shortened. Therefore, the FCM-MFOA-GRNN method is feasible and efficient for processing large data volumes in cloud computing environments.

6. Conclusion

Aiming at the poor detection performance of existing intrusion detection methods in the environment of high-dimensional massive data and uneven class distribution, an anomaly traffic detection method based on deep learning in cloud computing environment is proposed. By introducing the combination of the FCM algorithm and GRNN, the stability of anomaly traffic detection system is improved. Meanwhile, MFOA is used to optimize the parameters of the FCM-GRNN method to speed up the convergence. Experimental results show that the proposed method has good detection ability.

The experiment only considers one intrusion detection dataset, so more datasets can be involved to train the detection model in the future. Moreover, anomaly detection is only one aspect. How to take mitigation measures to reduce the damage caused by network attack is also a direction of great research value.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] Z. H. A. N. G. Yong-dong, C. H. E. N. Si-yang, P. E. N. G. Yu-he, and Y. A. N. G. Jian, "A survey of deep learning based network intrusion detection," *Journal of Guangzhou University Natural Science Edition*, vol. 18, no. 3, pp. 17–26, 2019.

[2] J. Huang, W. Zhang, W. Huang, W. Huang, L. Wang, and Y. Luo, "High-resolution fiber optic seismic sensor array for intrusion detection of subway tunnel," in *2018 Asia Communications and Photonics Conference (ACP)*, pp. 1–3, Hangzhou, China, October 2018.

[3] C. Deng and H. Qiao, "Network security intrusion detection system based on incremental improved convolutional neural network model," in *International Conference on Communication and Electronics Systems.*, pp. 1–5, Coimbatore, India, 2016.

[4] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *IEEE Communication Surveys and Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[5] N. El Moussaid and A. Toumanari, "Overview of intrusion detection using data-mining and the features selection," in *International Conference on Multimedia Computing and Systems*, pp. 1269–1273, Marrakech, Morocco, 2014.

[6] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, "Evaluating computer intrusion detection systems," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–41, 2015.

[7] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[8] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: experiments and analyses," *Pattern Recognition*, vol. 74, no. 4, pp. 406–421, 2018.

[9] Y. Mehmood, M. A. Shibli, U. Habiba, and R. Masood, "Intrusion detection system in cloud computing: challenges and opportunities," in *2nd National Conference on Information Assurance*, pp. 59–66, Rawalpindi, Pakistan, 2013.

[10] A. Drewek Ossowicka, M. Pietrołaj, and J. Rumiński, "A survey of neural networks usage for intrusion detection systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 497–514, 2021.

[11] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, no. 6, pp. 147–167, 2019.

[12] A. Bakshi and Sunanda, *A comparative analysis of different intrusion detection techniques in cloud computing*, Springer, Singapore, 2019.

[13] S. G. Kene and D. P. Theng, "A review on intrusion detection techniques for cloud computing and security challenges," in *2nd international conference on electronics and communication systems*, pp. 227–232, Coimbatore, India, 2015.

[14] N. Keegan, S. Y. Ji, A. Chaudhary, C. Concolato, B. Yu, and D. H. Jeong, "A survey of cloud-based network intrusion detection analysis," *Human-centric Computing and Information Sciences*, vol. 6, no. 1, pp. 1–16, 2016.

[15] A. Aburomman and M. B. I. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," *Computers & Security*, vol. 65, no. 4, pp. 135–152, 2017.

[16] A. Nisioti, A. Mylonas, P. D. Yoo, and V. Katos, "From intrusion detection to attacker attribution: a comprehensive survey of unsupervised methods," *IEEE Communication Surveys and Tutorials*, vol. 20, no. 4, pp. 3369–3388, 2018.

[17] L. N. Tidjon, M. Frappier, and A. Mammar, "Intrusion detection systems: a cross-domain overview," *IEEE Communication Surveys and Tutorials*, vol. 21, no. 4, pp. 3639–3681, 2019.

[18] C. Liu, J. Wang, J. Xu, J. Wang, C. Liu, and Y. Wang, "Abnormal data flow detection in the Internet of things," in *4th International Conference on Electronics and Communication Engineering*, Xi'an, China, 2021.

[19] M. Alauthman, N. Aslam, M. Al-Kasassbeh, S. Khan, A. Al-Qerem, and K. K. R. Choo, "An efficient reinforcement learning-based Botnet detection approach," *Journal of Network and Computer Applications*, vol. 150, no. 11, article 102479, 2020.

[20] W.-H. Lin, H.-C. Lin, P. Wang, B.-H. Wu, and J.-Y. Tsai, "Using convolutional neural networks to network intrusion detection for cyber threats," *international conference on applied system invention*, 2018, pp. 1107–1110, Chiba, Japan, 2018.

[21] R. Blanco, P. Malagón, J. J. Cilla, and J. M. Moya, "Multiclass network attack classifier using CNN tuned with genetic algorithms," in *28th international symposium on power and timing modeling, optimization and simulation*, pp. 177–182, Platja d'Aro, Spain, 2018.

[22] A. H. Mirza and S. Cosan, "Computer network intrusion detection using sequential LSTM neural networks autoencoders," *26th signal processing and communications applications conference*, 2018, pp. 1–4, Izmir, Turkey, 2018.

[23] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," in *20th international conference on advanced communication technology*, pp. 178–183, Chuncheon, Korea (South), 2018.

[24] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, no. 5, pp. 52843–52856, 2018.

[25] A. L. H. Muna, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial Internet of things based on deep learning models," *Journal of Information Security and Applications*, vol. 41, no. 12, pp. 1–11, 2018.

[26] R. C. Staudemeyer and C. W. Omlin, "Evaluating performance of long short-term memory recurrent neural networks on intrusion detection data," in *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, pp. 218–224, New York, NY, United States, 2013.

[27] Y. Zhang, X. Chen, D. Guo, M. Song, Y. Teng, and X. Wang, "PCCN: parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows," *IEEE Access*, vol. 7, no. 9, pp. 119904–119916, 2019.

[28] A. Pektaş and T. Acarman, "A deep learning method to detect network intrusion through flow-based features," *International Journal of Network Management*, vol. 29, no. 3, pp. 2019–2026, 2019.